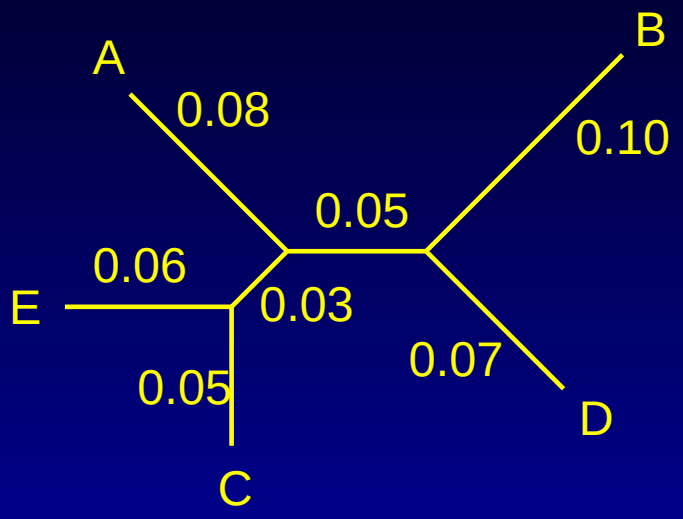
Lecture 23. Phylogeny methods, part 3 (Distance methods)

Joe Felsenstein

Department of Genome Sciences and Department of Biology

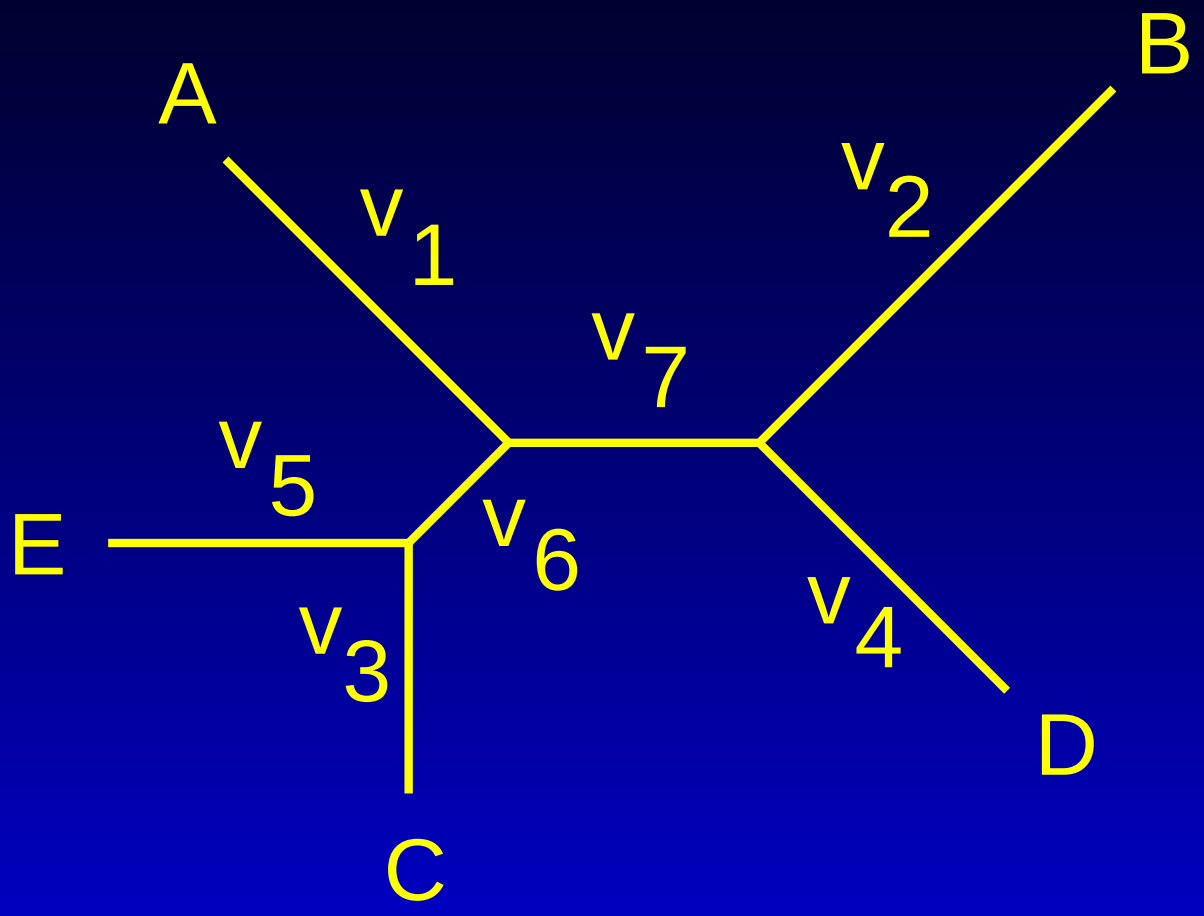
A phylogeny with branch lengths



	A	B	C	D	E
A	0	0.23	0.16	0.20	0.17
B	0.23	0	0.23	0.17	0.24
C	0.16	0.23	0	0.15	0.11
D	0.20	0.17	0.15	0	0.21
E	0.17	0.24	0.11	0.21	0

and the pairwise distances it predicts

A phylogeny with branch lengths



Least squares trees

Least squares methods minimize

$$Q = \sum_{i=1}^{n} \sum_{j \neq i} w_{ij} (D_{ij} - d_{ij})^2$$

over all trees, using the distances d_{ij} that they predict. Cavalli-Sforza and Edwards suggested $w_{ij} = 1$, Fitch and Margoliash suggested $w_{ij} = 1/D_{ij}^2$.

Statistical assumptions of least squares trees

Implicit assumption is that distances are (independently?) Normally distributed with expectation d_{ij} and variance proportional to $1/w_{ij}^2$:

$$D_{ij} \sim \mathcal{N}(d_{ij}, K/w_{ij})$$

Thus the different weightings correspond to different assumptions about the error in the distances. Also, there is assumed to be no covariance of distances.

In fact, the distances will covary, since a change in an interior branch of the tree increases (or decreases) all distances whose paths go through that branch.

Matrix approach to fitting branch lengths

If we stack the distances up into a column vector $\mathbf{D}$, we can solve the least squares equation (obtained by taking derivatives of the quadratic form Q):

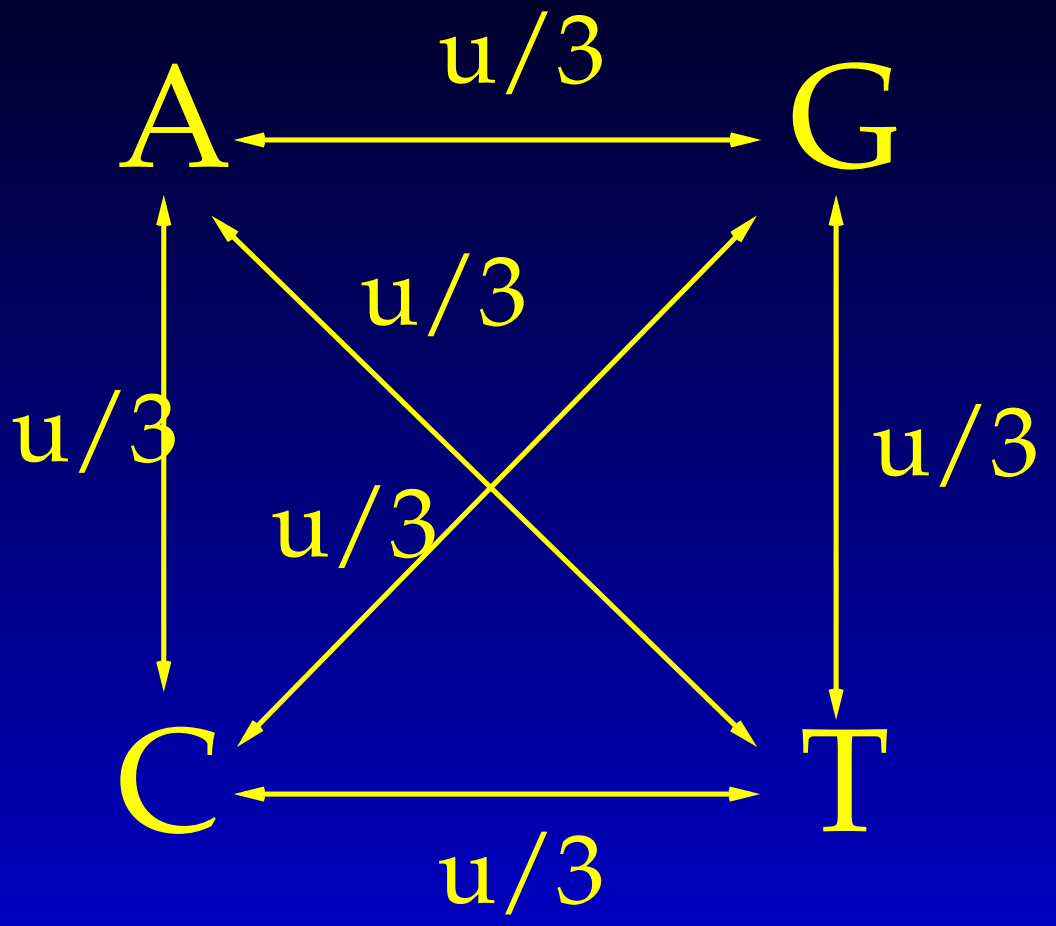
$$\mathbf{D}^{\mathsf{T}} = (D_{12}, D_{13}, D_{14}, D_{15}, D_{23}, D_{24}, D_{25}, D_{34}, D_{35}, D_{45})$$

$$\mathbf{X}^{\mathsf{T}}\mathbf{D} = \left(\mathbf{X}^{\mathsf{T}}\mathbf{X}\right)\mathbf{v}.$$

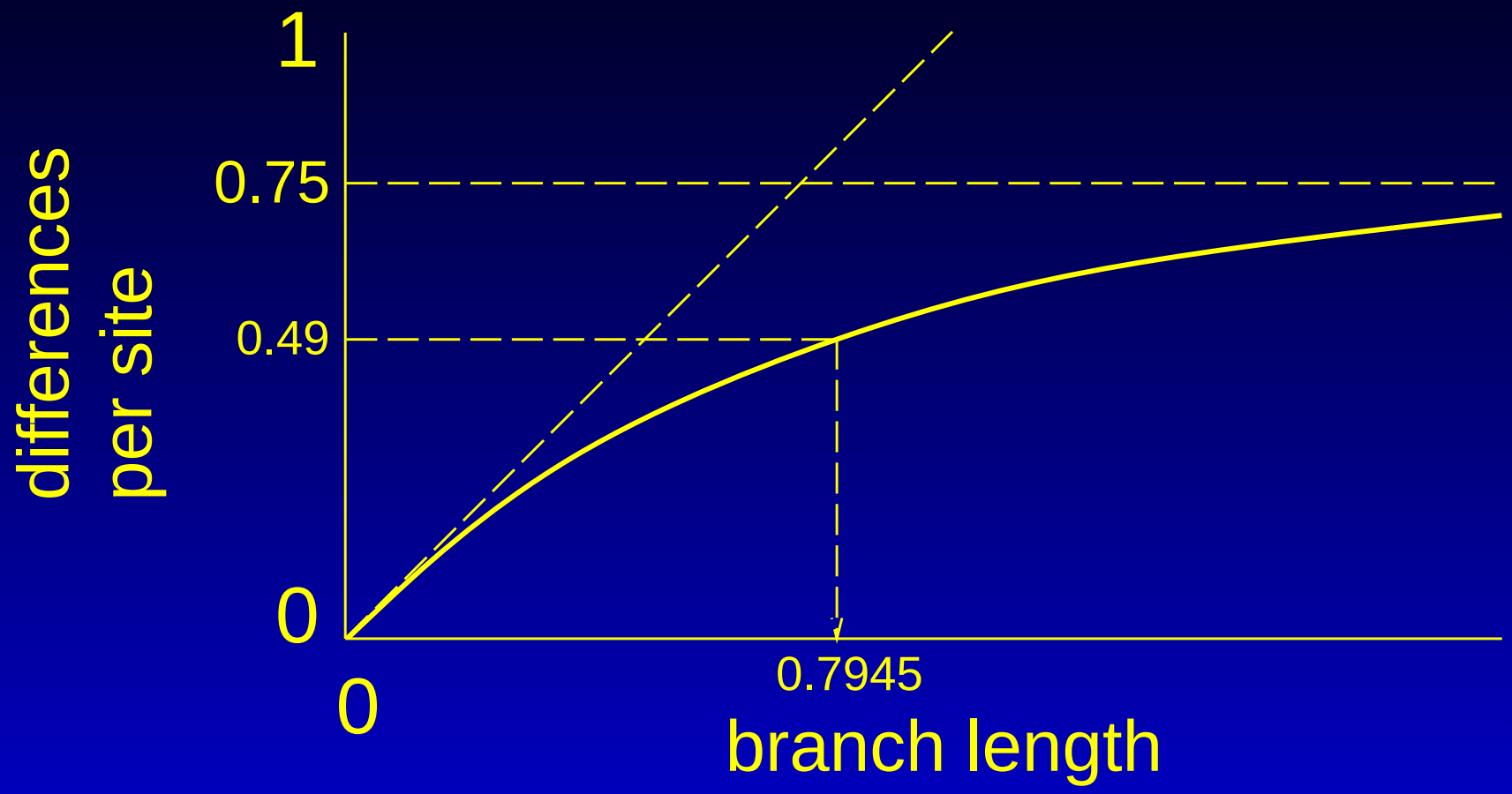
where the “design matrix" $\mathbf{X}$ is has 1’s whenever a given branch lies on the path for the given distance.

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

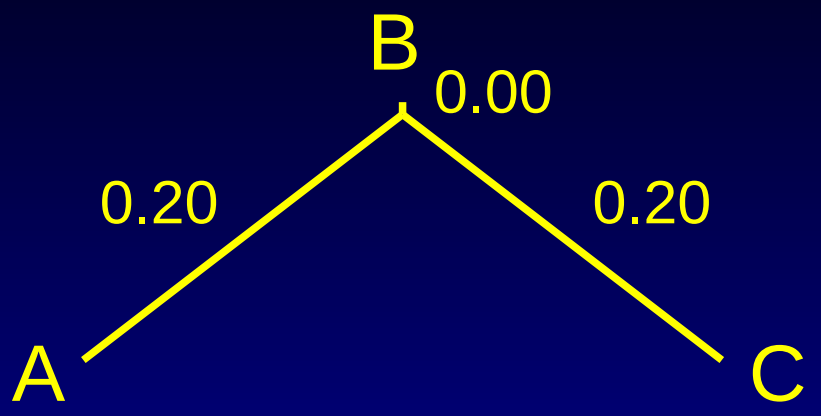
The Jukes-Cantor model for DNA

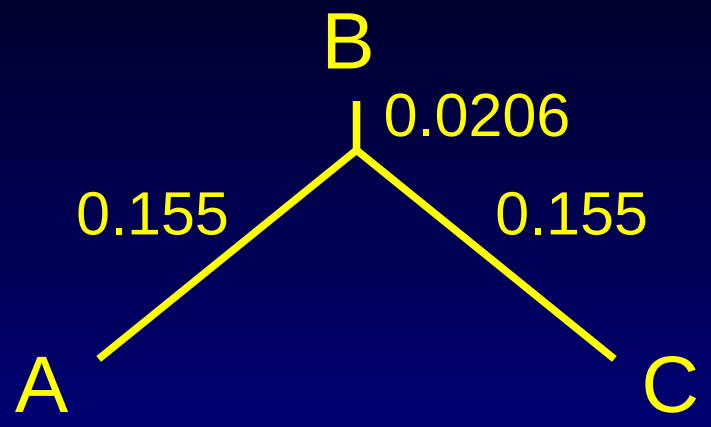


The distance for the Jukes-Cantor model



If you don’t correct for “multiple hits"





Left: the true tree. Right: a tree fitting the uncorrected distances

Approximate variances for distances

under the Jukes-Cantor model

Distance as a function of fraction of nucleotide differences is

$$\hat{t} = -\frac{3}{4}\ln\left(1 - \frac{4}{3}D\right)$$

The “delta method" approximates the variance of one as a function of the variance of the other:

$$\mathrm{Var}(\hat{t}) \simeq \left(\frac{\partial \hat{t}}{\partial D}\right)^2 \mathrm{Var}(D)$$

Approximate variances, continued

The variance of fraction of nucleotide difference with n sites is the binomial variance

$$\mathrm{Var}(\mathsf{D}) \;=\; \mathsf{D}\,(1-\mathsf{D})/\mathsf{n}$$

and since

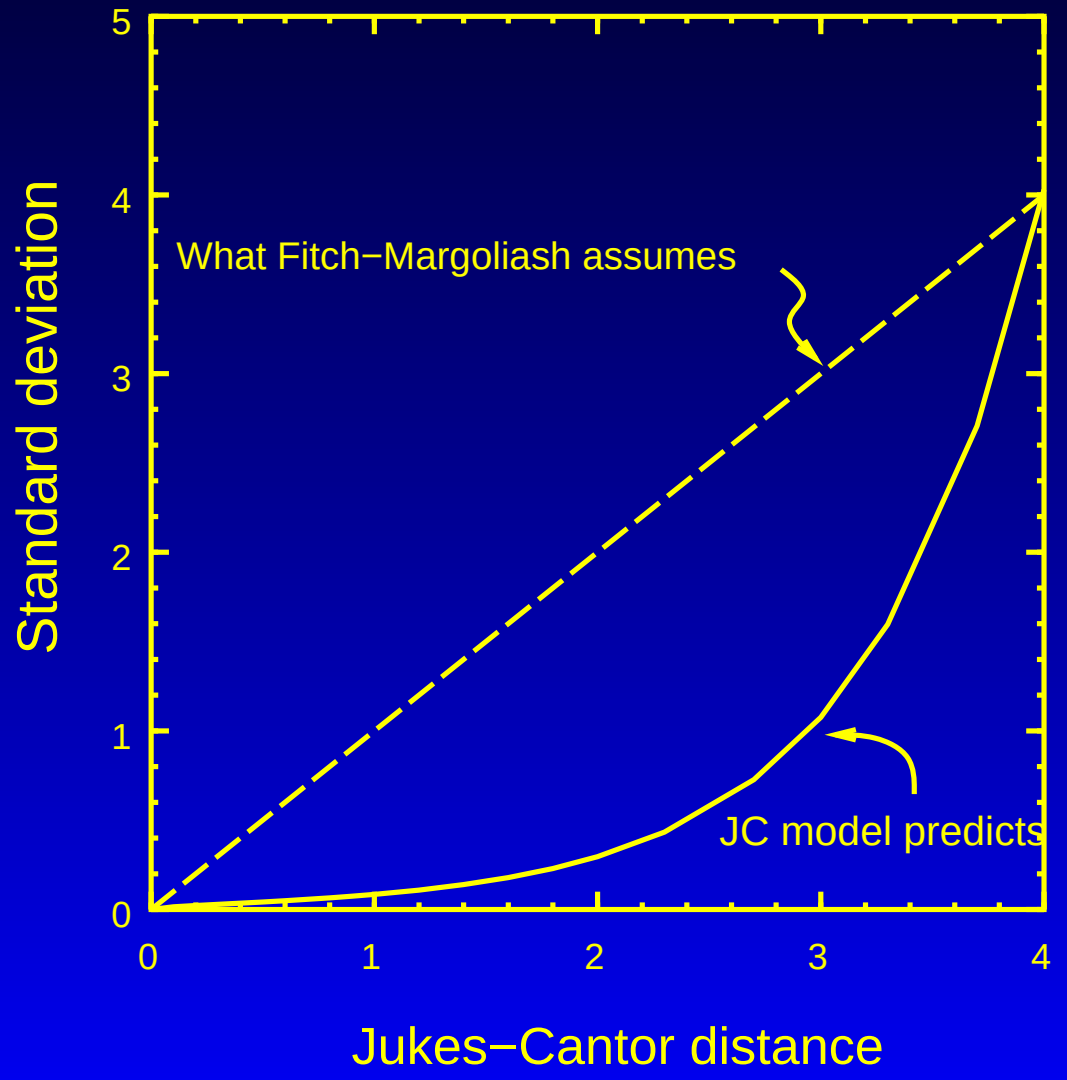
$$\frac{\partial \hat{\mathsf{t}}}{\partial \mathsf{D}} \;=\; \frac{1}{1-\frac{4}{3}\mathsf{D}}$$

we get

$$\mathrm{Var}(\mathsf{D}) \;\simeq\; \frac{\mathsf{D}(1-\mathsf{D})/\mathsf{n}}{\left(1-\frac{4}{3}\mathsf{D}\right)^2}$$

Standard deviation of distance

as it increases with distance (given the JC model)



The UPGMA algorithm

1. Choose the smallest of the D_{ij}
2. make a new "tip" (ij)
3. Have i and j connected to this new tip, by a node whose "time" ago in branch length units is $D_{ij}/2$.
4. Have the weight of the new tip be $w_{(ij)} = w_i + w_j$
5. For each other tip, aside from i and j, compute

$$D_{(ij),k} = D_{k,(ij)} = \frac{w_i D_{ik} + w_j D_{jk}}{w_i + w_j}$$

6. Delete the rows and columns of the D matrix for i and j.
7. If only one row left, stop, else return to step 1.

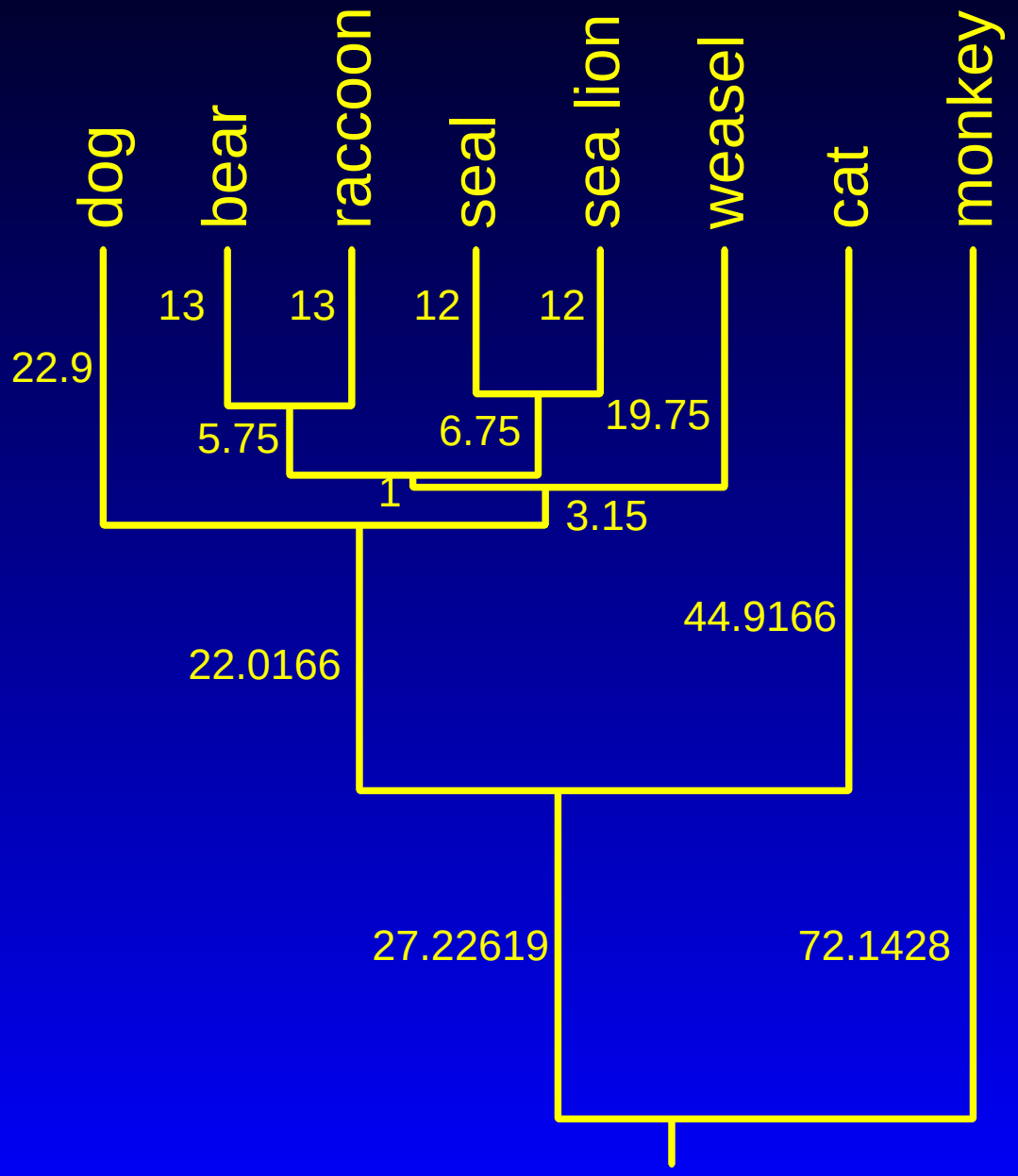
This can be done in $O(n^2)$ time if you save minimum elements of each row.

Sarich's (1969) immunological distances

with columns and rows corresponding to the smallest distance highlighted.

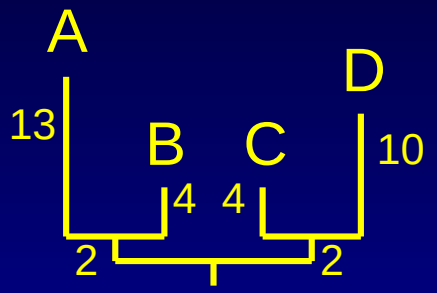
	dog	bear	raccoon	weasel	**seal**	**sea lion**	cat	monkey
dog	0	32	48	51	**50**	**48**	98	148
bear	32	0	26	34	**29**	**33**	84	136
raccoon	48	26	0	42	**44**	**44**	92	152
weasel	51	34	42	0	**44**	**38**	86	142
seal	**50**	**29**	**44**	**44**	**0**	**24**	**89**	**142**
sea lion	**48**	**33**	**44**	**38**	**24**	**0**	**90**	**142**
cat	98	84	92	86	**89**	**90**	0	148
monkey	148	136	152	142	**142**	**142**	148	0

UPGMA tree for Sarich (1969) data



UPGMA misleads on a nonclocklike tree

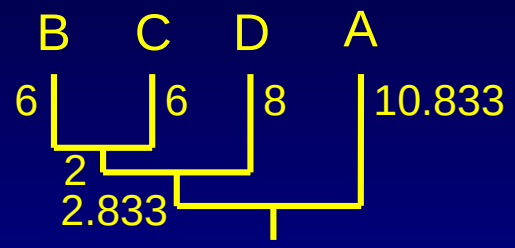
True tree



Distance matrix

	A	B	C	D
A	0	17	21	27
B	17	0	12	18
C	21	12	0	14
D	27	18	14	0

UPGMA tree



An unclocklike tree (left), the distances from it (center) and the UPGMA tree from those distances (right)

The distortion of the tree is due to “short-branch attraction" in which B and C, close to each other in the true tree, cluster first.

Neighbor-joining algorithm

1. For each tip, compute $u_i = \sum_{j \neq i}^{n} D_{ij}/(n-2)$
2. Choose the i and j for which $D_{ij} - u_i - u_j$ is smallest.
3. Join items i and j. Compute the branch length from i to the new node (v_i) and from j to the new node (v_j) as

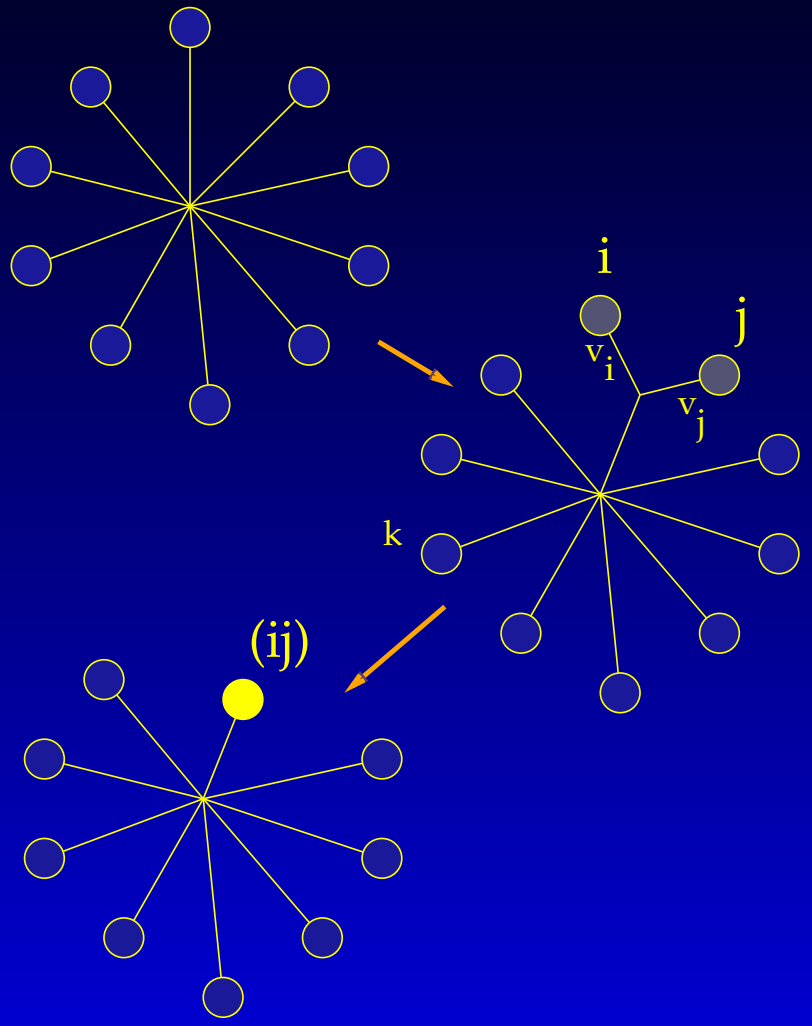
$$\begin{aligned} v_i &= \frac{1}{2}D_{ij} + \frac{1}{2}(u_i - u_j) \\ v_j &= \frac{1}{2}D_{ij} + \frac{1}{2}(u_j - u_i) \end{aligned}$$

4. compute the distance between the new node (ij) and each other tip as

$$D_{(ij),k} = (D_{ik} + D_{jk} - D_{ij}) \Big/ 2$$

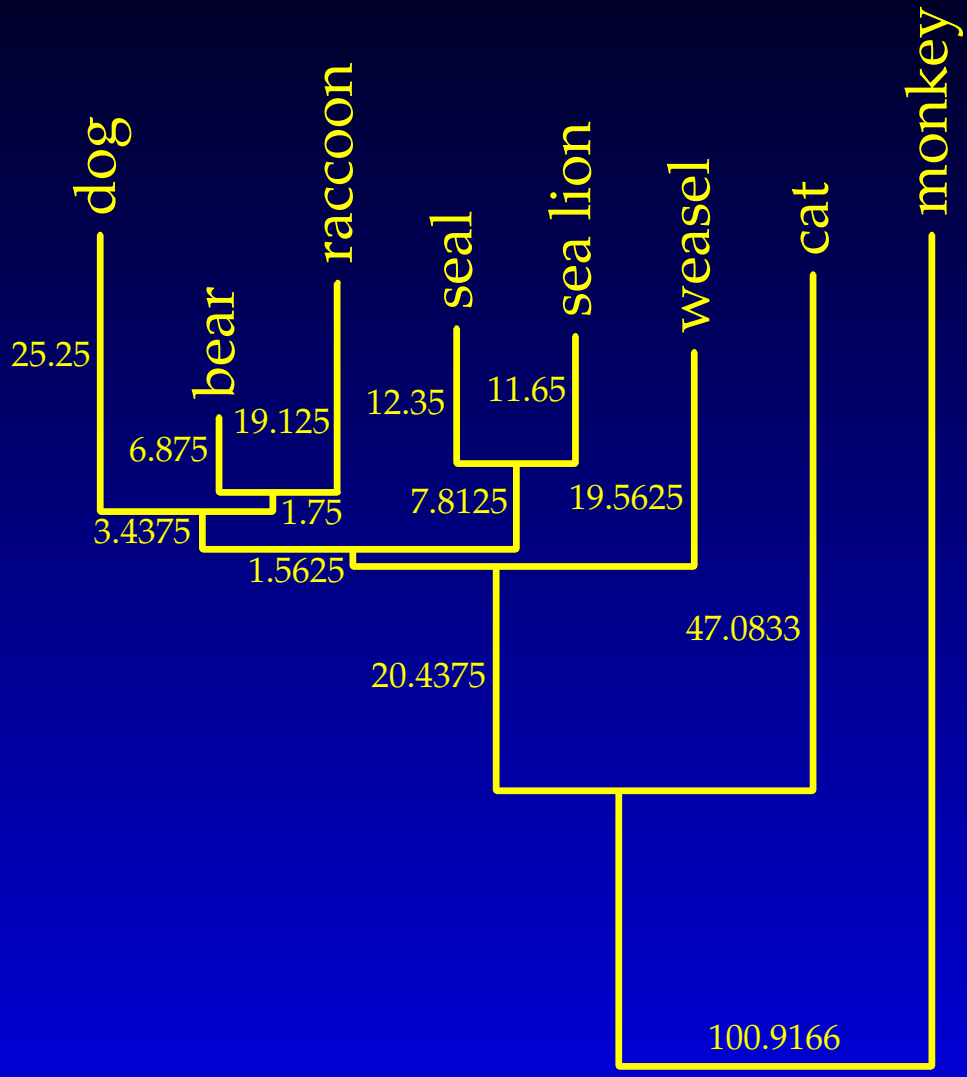
5. delete tips i and j from the tables and replace them by the new node, (ij), which is now treated as a tip.
6. If more than two nodes remain, go back to step 1. Otherwise connect the two remaining nodes by a branch of length D_{ij}.

Star decomposition search



"Star decomposition" tree search method used in Neighbor-Joining method

NJ tree for Sarich's (1969) data



Neighbor-joining tree for the Sarich (1969) imunological distance data

References, page 1

Bryant, D., and P. Waddell. 1998. Rapid evaluation of least-squares and minimum-evolution criteria on phylogenetic trees. *Molecular Biology and Evolution* **15:** 1346-1359. **[quicker least squares distance trees]**

Bruno, W. J., N. D. Socci, and A. L. Halpern. 2000. Weighted neighbor joining: a likelihood-based approach to distance-based phylogeny reconstruction. *Molecular Biology and Evolution* **17:** 189-197. **[A weighted version of NJ which de-weights large distances appropriately]**

References, page 1

Cavalli-Sforza, L. L., Edwards, A. W. F. 1967. Phylogenetic analysis: models and estimation procedures. *Evolution* **32:** 550-570 (also published in *American Journal of Human Genetics* **19**: 233-257, 1967) **[One of the first least squares distance methods]**

Farris, J. S. 1981. Distance data in phylogenetic analysis. pp. 3-23 in *Advances in Cladistics. Proceedings of the first meeting of the Willi Hennig Society.*, ed. V. A. Funk and D. R. Brooks. New York Botanical Garden, Bronx. **[Criticism of distance methods]**

Farris, J. S. 1985. Distance data revisited. *Cladistics* **1:** 67 -85. **[Reply to my 1984 paper]**

Farris, J. S. 1986. Distances and statistics. *Cladistics* **2:** 1 44-157. **[debate was cut off after this]**

References, page 1

Felsenstein, J. 1984. Distance methods for inferring phylogenies: a justification. *Evolution* **38:** 16-24. **[Argument for statistical interpretation of distance methods]**

Felsenstein, J. 1986. Distance methods: reply to Farris. *Cladistics* **2:** 130-143. **[reply to Farris 1985]**

Fitch, W. M. and E. Margoliash. 1967. Construction of phylogenetic trees. *Science* **155:** 279-284. **[One of the first least squares distance methods]**

Michener, C. D. and R. R. Sokal. 1957. A quantitative approach to a problem in classification. *Evolution* **11:** 130-162. **[UPGMA]**

Saitou, N., and M. Nei. 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution* **4:** 406-425. **[Neighbor-joining]**

How it was done

This projection produced

- using the `prosper` style in LaTeX,
- using Latex to make a `.dvi` file,
- using `dvips` to turn this into a Postscript file,
- using `ps2pdf` to mill it into a PDF file, and
- displaying the slides in Adobe Acrobat Reader.

Result: nice slides using freeware.